"MOM! The Vacuum Cleaner is Chasing the Dog Again!"

J. P. Gunderson and L. F. Gunderson

Gamma Two, Inc.
Denver, CO 80223

ABSTRACT

The consistent drive to make everything from toasters to autonomous combat aircraft 'smart' forces an examination of the cost-effectiveness of intelligence, as well as its general applicability. In this paper we draw on examples from natural intelligent systems and suggest that it is possible to make intelligent systems too smart. We suggest that there is an 'appropriate inteligence' for all tasks, and that adding more intelligence may have unexpected negative consequences.

KEYWORDS: *Artificial Intelligence, Robotics, Measuring Intelligence, Appropriate Intelligence*

1 INTRODUCTION

There is a trend in recent artificial intelligence research to put more and more intelligence into just about everything. This runs the gamut from intelligent 'internet-aware' toasters, to land-mine detection robots, to computer viruses that adapt to the security installed on individual machines. This push towards smarter and smarter devices brings up the question "When is too much intelligence a bad thing?"

In this paper we present a few arguments and suggest the use of 'appropriate intelligence' for the tasks and responsibilities of an intelligent system. In addition, we propose that too much intelligence may be worse than insufficient intelligence in some task domains.

The domain that we use as an example is that of a lowly service robot (See Figure 1), that thankless drudge that rolls up and down countless hallways in hotels and office buildings cleaning the carpets; or comes out when the house is empty to clean up the debris from last night's cocktail party - the vacuum cleaning robot. In a 1995 study, the commercial cleaning market was estimated to be worth approximately $40 Billion[1], and the obstacle that was identified as preventing greater use of automated cleaning robots was insufficient processing power. In short, they just are not smart enough. In this paper, we discuss some of the potential consequences of "just make them smarter."

Donald Knuth was quoted as saying that "Science is what we understand well enough to explain to a computer. Art is everything else we do." We don't really know how to make smart machines, and there are suggestions that we don't really know what 'smart' is[2]. If we can't define it, how can we begin to explain it to a computer? While much progress has been made over the last forty years, one common thread has always been that "We don't have sufficient processing power." When machines were equiped with Megahertz processors, and Megabyte RAMs, the argument was that we really needed Gigahertz machines with Gigabyte RAM. Now that such computers sell for under $1000 at any electronics store, the argument is that we need Terahertz machines with Terabyte RAM. Perhaps the problem is not that the machines aren't capable, but that we don't know how to tell them to be smart.

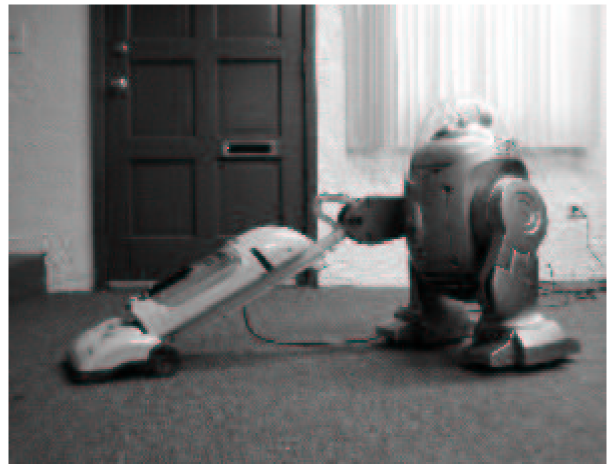

Fig. 1. A prototypical service robot. This is a Robo-Scout impressed into cleanup duty, using a Eureka bagless vacuum cleaner.

But, even if we knew what intelligence was, should we make every machine as smart as possible? As humans, we are an example of a nominally intelligent entity. Why isn't every species equally intelligent?

This argument has two parts:

1. Intelligence is expensive, in both software and wetware (biological intelligence).
2. The deployment of excessive intelligence for a specific task may have serious unintended consequences.

In this paper we will use a simple definition of an intelligent system - one that can overcome obstacles to achieve its goals. This is a blending of the problem solving approach suggested by Newell and Simon[3] and the reactive systems approach used by many robotics researchers. In short, we know something is intelligent by how it responds when things go wrong. If every attempt at goal achievement runs perfectly smoothly, there may be no intelligence at all, but if we see the system attempt a goal, fail, overcome these obstacles, and then succeed, we would be tempted to argue that it is intelligent. As Edwin Boring put it "There is little that will make our robot seem more human than this ability to choose one means after another until the goal is reached."

2 INTELLIGENCE IS EXPENSIVE

It is hard to make anything behave intelligently. Certainly, significant strides have been made in embedding rudimentary intelligence into some systems, but designing, developing, and deploying something that acts intelligently consumes major resources on the development side, and seems to require extreme amounts of computational resources on the deployment side.

2.1 *Biological Intelligence*

Using biological systems as perhaps the only exemplars of deployed, long-term intelligent systems, it is clear that there is some sort of a trade-off between the cost of developing and maintaining intelligence, and the benefits that this intelligence provides to the species. While any individual mouse might prosper from being able to outsmart a cat, clearly, the costs have outweighed the benefits for most species of mice.

Biological organisms can be viewed as very highly optimized systems, with very few slack resources. If resources are used to create and maintain a feature that is not necessary for survival, then those resources cannot be used for any other purpose. As a result, another type of organism could fill the same ecological niche, consuming fewer resources, and thus be better fitted to survive[4]. This applies to gaudy displays of feathers, bioluminescent lures, and to brains. In fact, brains are very expensive investments. In humans the brain makes up approximately 2% of the body mass, yet uses 20% of the calories ingested[5]. Gram for gram, the brain requires ten times as much energy as the body as a whole: brains are very expensive components to maintain in a biological organism.

Let us look at a task that we can acheive with a robotting system, foraging and caching. Imagine a small robot trash collector. It travels around a space collecting litter, and transporting that litter to trach cans. To do so, it uses vision to locate the litter, localizes itself in the world, and then travels to a memorized location (using some form of waypoint navigation) to drop off the trash. This behavior is analogous to food caching behavior demonstrated by several species, with one critical difference. If the animal cannot find the seed caches, it will starve to death. Presumably, the more food it can cache (and find) the better its chances for survival.

The cognitive load is low if the bird places all the seeds in one cache, but if that single cache is destroyed, the results are deadly. So, many bird species have developed the strategy of creating a large number of small food caches, and then returning to these caches to recover the stored food. There is a significant (for a bird, anyway) cognitive load associated with managing a large number of caches. The individual cache locations must be memorized (or some algorithmic mechanism must be used to 'rediscover' them), and the bird must also remember which caches have been emptied and which still retain food. In birds, the hippocampus is critical for spatial memory[6], and increased spatial memory corresponds to increased size of the hippocampus.

Beyond the basic idea that there is no such thing as a free lunch, is there any evidence that there are tradeoffs between increasing intelligence versus improving other characteristics that lead to higher success of an individual or a species? If there is no trade-off, we would expect to see the same types of cognitive fuction in closely related species. Conversely, if there is a trade-off, then we would expect to see closely related species with different cognitive loads, developing different cognitive abilities.

Three closely related bird species, all living in the same ecosystem, have been studied to determine the relationship between cognitive need and cognitive function [7]. These three species are:

1. Clark's nutcracker (*Nucifraga columbiana*),
2. the pinyon jay (*Gymnorhinus cyanocephalus*), and
3. the scrub jay (*Aphelocoma coerulescens*).

While all three of these birds cache seeds, the Clark's nutcracker is the most prodigious seed gatherer. This species can cache up to 35,000 seeds in individual caches of approximately 15 seeds each[8]. The other two corvids also cache seeds, but in lesser numbers. The ability to locate these caches is regulated by the hippocampus. If there was no tradeoff to the growth of the hippocampus then we would expect that all three of these species would have a similarly large hippocampus. However, the size of the hippocampus is proportional to the seed caching ability of the species[9].

In addition, one of the three corvids, the pinyon jay, is quite social, while the scrub jay is solitary. Testing was done on these two species, to assess their ability in complex cognitive tasks of the type that would be relevant

Fig. 2. Clark's nutcracker, This small bird stores seeds in up to 3000 individual caches, and uses waypoints to navigate back to the storage locations. To achieve this, the bird's hippocampus is increased in size when compared with related birds in the same ecosystem.

to their ability to assess and track social relationships. In one of these, the ability of the bird to infer the ordering of a set of arbitrary relationships from a series of dyadic comparisons was tested. In the second, their ability to make transitive inferences was tested. The pinyon jays scored higher on both types of tests[10].

These three species have distinct techniques for surviving in very similar ecological niches. One has developed a larger hippocampus to support the cognitive load of cache management, one has developed increased social network skills, and the third has neither of these. Each is surviving and competing in the same environment. However, none of these has both the increased hippocampus and the increased networking skills. One would presume that such a bird would have the benefits of both the Clark's nutcracker and the pinyon jay, and would out compete both of the others. We posit that the cost of building and maintaining a more complex brain is the limiting factor.

2.2 *Machine Intelligence*

Embedding intelligence into a machine is also very expensive, both in building (designing and developing) the intelligence, and in maintaining the computational resources needed to act intelligently. We will approach this from two ends of the machine intelligence spectrum: the costs of deliberative intelligence, and the costs of reactive intelligence.

Deliberative intelligent systems are fairly easy to develop. The engine needed to do a brute force exploration of possible world states is a well known, solved problem. The development cost of a deliberative system is quite reasonable, provided that one only wants to work on toy problems. Making a deliberative system that is capable of producing solutions to more complex, real world problems; and producing these solutions in a reasonable time frame becomes more challenging. The cost of a solution is exponential in the number of factors in the domain, and exponential in the depth of the planning process. In addition, since these systems are based in an internal model of the domain, it is fairly expensive to deploy a deliberative system into a domain. However, the major problem seems to be actually using the deliberative system to achieve goals in the real world.

It has been well established that the model of deliberative cognition is computationally expensive. This has been shown in domains such as robotic systems [11, 12], planning systems[13–16], and two player game playing[17]. The general approach is the further the system can 'look ahead' the more likely it is to meet its goals. However, the greater the look ahead, the exponentially greater the computation; the greater the resolution of the internal model, the greater the maintenance costs of keeping the intelligent system current. In response to these problems, reactive systems were proposed.

Reactive systems have been defined as systems with tightly coupled perception and action, typically in the context of motor behaviors, which produce timely robotic response in dynamic and unstructured worlds[18]. In effect, the system is designed so that whenever a significant sensory input is detected, it responds with the 'correct' action/output. Frequently modeled using a subsumption architecture[19], these systems are easy to deploy, and can exhibit complex behaviors that seem intelligent. However, it can be devilishly difficult to design the behaviors to achieve the 'correct behavior'. Researchers[20] have done comparative analysis of the behavior of reactive systems and humans suffering from Korsakov syndrome[1] They concluded that, even with reactive system, there is a tradeoff between the system getting stuck in cyclical behavior and the ability of the system to perform complex behaviors. In effect, adding additional layers to the layered control system resulted in un-intended consequences that prevented the system from behaving intelligently.

It seems that just like in biological systems, intelligence is very expensive, either in designing a reactive system to achieve goals, or in the computational burden of deploying a deliberative system into the real-world. Of course, most deployed intelligent systems are neither purely reactive nor purely deliberative. The use of hybrid systems is the norm, and numerous hybrid architectures have been proposed and are under development[21, 11, 22].

But, let as assume that we have the ability to put as much intelligence as we wish into a deployed system. When we look at computational systems it is also necessary to balance the benefits of increased intelligence. Hav-

[1] A medical condition that results in the loss of memory of recent events, while impressions of long ago are recalled properly.

ing a 100 Gigabit per second processor and a Gigabyte RAM (Hans Moravec's estimate of a mouse brain[23]) in a vacuum cleaner may not result in an appreciably cleaner floor than that provided by a system using a random walk circuit.

3 TOO MUCH INTELLIGENCE?

But suppose the customers were willing to pay for the development and deployment, or suppose that we suddenly had the positronic brain that allowed us to put as much intelligence as we wanted into any device that we chose. Surely nothing could go wrong by making the intelligent system as smart as possible, could it? This may depend on one's point of view. Certainly this scenario has been presented in numerous ways from Capeks "Rossum's Universal Robots"[24]" through Authur C. Clarke's "2001: A Space Odyssey". The notion of an intelligent system that is free to make its own decisions has been, and continues to be, the focus of numerous workshops and conferences.

One issue is the notion of how smart should a system be allowed to be. The simple answer of 'as smart as possible' has led to some interesting conclusions. We, as the designers and employers of these intelligent tools, want them to have several characteristics such as:

- Robustness and Fault Tolerance,
- Reliability,
- Flexibility and Adaptability, and
- Coherence[25].

Yet these design goals contain within them internal conflicts. It is a common experience to test an intelligent system and during testing discover that the system has achieved its goals, but in a way that was unintended by the designer. Sometimes the system produces a high quality solution, but far more often the solution is unacceptable. The system was intelligent enough to see a solution that took advantage of a 'loophole' in the problem specification, causing the designer to enter into a battle of wits - trying to specify the problem precisely enough to allow the system to find the 'right' solution, but not so tightly that robustness and flexibility are sacrificed in pursuit of reliability. During early work on a control system for a maintenance robot, the planning system 'discovered' that the easiest way to effect a repair of a broken machine was to strip working parts off a functioning instrument[26]. Then, when instructed to fix this newly broken machine, all it had to do was replace the parts that it had removed earlier, thus ensuring perpetual employment for the robot. In one sense this is a very intelligent solution, but not the kind of solution we want to achieve.

Imagine a really intelligent vacuum cleaner, a task oriented, goal driven, autonomous vacuum cleaner. It has a job to do - keep the carpets clean in the family room. If it were just a dumb vacuum cleaner it would simply bounce around the floor at random, sucking up dirt, and avoiding obstacles[27].

But this wastes power, since it is often cleaning sections that are not dirty; and it is inefficient, since it may cover the center of the room far more often than the corners which are dirty. So let's give it more intelligence. Now it can estimate the traffic in the room, and can localize itself, so it covers the floor more efficiently, and only vacuums when the room has been used[28]. So far, so good. But - surely we can do better, let's make it even smarter.

4 AS SMART AS POSSIBLE

Let's embed some significant intelligence into our robot, the ability to learn the habits of the people with whom it shares a workspace, the ability to learn that when Susie uses the room there are potato chips all over the floor, and she drops those hair pins that cause down time. When Dave uses the room to study there is no point coming out to vacuum, because there is nothing to clean up. But worst of all is the poodle Fifi. She comes in after running through the woods, and lies on the carpet and chews the dried mud out of her fur, and works those burrs out and spits them on the carpet.

In fact, in applying a little reinforcement learning to the problem, the vacuum cleaner could easily produce the following analysis: "It is clear that 60 to 70 percent of my workload is generated every time that dog comes into the family room. So, given the task of keeping the carpet clean, and managing energy costs, it would be best if that dog never came into the room at all. Hmmmmm, maybe a little experimentation is in order..." This might result in the following dialog:

> "Mom!, the vacuum cleaner is chasing the dog again!"
>
> "David, you're an engineer, can't you do something with that machine?"
>
> "Let's just alter your programming a little...."
>
> "I'm sorry Dave, I'm afraid I can't do that."

Unfortunately, one of the unintended side effects of adding unneeded complexity to artifacts is the need to control the interactions between various sub-components. Since the number of possible interactions increases factorially with the number of components, we quickly get to the point where the side effects cannot be controlled[29].

This type of problem has been explored in many complex systems. Several researchers[30, 31] suggest that as systems become more complex these internal conflicts will inevitably result in outcomes which are clearly wrong, which were unexpected until they occurred, but were inevitable in hindsight. Charles Perrow[32] calls these events *normal accidents*, and classifies systems according to the criteria of:

1. *interactions* - linear to complex, and
2. *coupling* - loose to tight.

According to Perrow, a linear system is one that has a general linear flow, no significant loops or feedback mechanisms. A system with complex interations has significant feedback, where the output of one stage of the process changes the inputs to an earlier stage, changing the outputs, ad infinitum. A system with tight coupling is one in which it is difficult, or impossible, to stop the process without causing problems, and one in which everything must flow in one of a few allowed pathways if the system is to function correctly. Based on these criteria, he identifies four classes of systems, and argues that tightly coupled systems with complex interactions are the most prone to inevitable failure.

Unfortunately, this seems to define an intelligent system, at least one that interacts with the world to achieve goals. Perrow's conclusions are that such systems need to be used sparingly, and only if necessary; and, if possible, the complexity and coupling of the systems need to be reduced to a minimum. In effect, the machines need to be just 'smart' enough to get the job done, and no smarter. Adding unnecessary intelligence will increase the risks of unexpected and unacceptable outcomes, increase the costs of designing and deploying the system, and provide no real benefits.

5 CONCLUSIONS

There are costs to adding more 'intelligence' to robots. Some of these costs are economic, some are structural, and some, inevitably, will be social. The economic costs of adding intelligence are dropping, along with the falling costs of hardware, and the growing sets of software tools to support intelligent systems. However, the structural costs have yet to be really experienced in the intelligent systems domain. So the push is towards putting more and more smarts into any possible system, just to get it to do something useful. This is a good thing on many levels, but perhaps all too quickly we will find that these intelligent machines don't look at the problem space in the same way we do, and that some of their intelligent solutions will be optimal for them, but sub-optimal for us.

This suggests that along with performance metrics for intelligent systems, we need to develop intelligence thresholds for mission tasking: give me a system that is just smart enough to get the job done, and not so smart it will screw the job up. The appropriate level of intelligence is the minimum amount needed for a specific system to reliably achieve a specific set of goals.

6 REFERENCES

1. Miller and Associates. The service robot market. Technical report, International Service robot Association, 1995.
2. K. Richardson. *The Making of Intelligence*. Columbia University Press, 2000.
3. Allen Newell and Herbert A. Simon. *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, USA, 1972.
4. N. A. Campbell. *Biology*, chapter 21: How Populations Evolve, pages 445–451. The Benjamin/Cummings Publishing Co., 1987.
5. M. E. Raichle and D. A. Gusnard. Appraising the brain's energy budget. *PNAS*, 99(16):10237–10239, August 6 2002.
6. E. M. Macphail. The role of the avian hippocampus in spatial memory. *Psicologica*, 23:93–108, 2002.
7. S. I. Yoerg. *Clever as as Fox: Animal Intelligence and What It Can Teach Us About Ourselves.* Bloomsbury: New York and London, 2001.
8. D. F. Tomback. Clark's nutcracker. *The Birds of North America*, (No 331), 1998.
9. S.D. Healy and J. R. Krebs. Food storing and the hippocampus in corvids: Amount and volume are correlated. *Proceedings of the Royal Society, London*, B23:241–245, 1992.
10. A. B. Kamil and R. P. Balda. Social complexity and transitiveinference in corvids. *Animal Behavior*, In Press, 2003.
11. R. P. Goldman, D. J. Musliner, M. S. Boddy, and K. D. Krebsbach. The circa model of planning and execution. In *Working Notes of the AAAI workshop on robots, softbots and immobots: Theories of action, Planning, and Control*, july 1997.
12. J. P. Gunderson and W. N. Martin. Effective planning horizons for robust autonomy. In *American Association for Artificial Intelligence Spring Symposium Series*. American Association for Artificial Intelligence, 2001.
13. J. Goldsmith, M. L. Littman, and M. Mundhenk. The complexity of plan existence and evaluation in probabilistic domains. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–97)*, pages 182–189, 1997.

14. T. Bylander. The computational complexity of propositional strips planning. *Artificial Intelligence*, 69:165–204, 1994.
15. K. Erol, J. Hendler, and D. S. Nau. HTN planning: Complexity and expressivity. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, volume 2, pages 1123–1128, Seattle, Washington, USA, 1994. AAAI Press/MIT Press.
16. N. Gupta and D. S. Nay. On the complexity of blocks-world planning. *Artificial Intelligence*, 56(2–3):223–254, August 1992.
17. D. Michie. Brute force in chess and science. In T. A. Marsland and J. Schaeffer, editors, *Computers, Chess, and Cognition*. Springer-Verlag, 1990.
18. R. Arkin. *Behavior–Based Robotics*. MIT Press, 1998.
19. R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14 – 23, March 1986.
20. A. Mukerjee and A. D. Mali. Reactive robots and amnesics: A comparative study in memoryless behavior. *IEEE/SMC Part C: Applications and Reviews*, 29(2):216–226, 1999.
21. J. S. Albus and A. M. Meystel. *Intelligent Systems: Architecture, Design, and Control*, chapter 1. John Wiley and Sons, 2002.
22. R. P. Bonasso and D. Kortenkamp. Characterizing an architecture for intelligent, reactive systems. In *AAAI-95 Spring Symposium*, 1995.
23. H. Moravec. *Mind Children*, chapter 2, pages 51–74. Harvard University Press, 1988.
24. K. Capek. *R. U. R. (Rossum's Universal Robots)*. Doubleday Page and Company, 1925.
25. *Generating Self-Reliant Teams of Autonomous Cooperating Robots: Desired Design Characteristics*, 1999.
26. J. P. Gunderson and W. N. Martin. The effects of uncertainty on plan success in a simulated maintenance robot domain. *Journal of Experimental and Theoretical Artificial Intelligence*, 12(2):153–164, 2000.
27. iRobot. Roomba intelligent vacuum instruction manual, 2003.
28. I. R. Ulrich, F. Mondada, and J. D. Nicoud. Autonomous vacuum cleaner. *Robotics and Autonomous Systems*, 19:233–245, 1997.
29. E. Tenner. *Why things bite back: Technology and the revenge of unintended consequences*. Vintage Books, 1997.
30. T. Kletz. *Learning from Accidents*. Gulf Professional Publishing, 2001.
31. J. Reason. *Human Error*, chapter 1, pages 7–9. Cambridge University Press, 1990.
32. C. Perrow. *Normal Accidents*, chapter 3, pages 96–105. Princeton University Press, 1999.